Bundit Lakhanugit

SOFT MAIL

การหาค่าสูงสุดของฟังก์ชันใด ๆ โดยใช้ Genetic Algorithm

Genetic Algorithm ทางเลือกหนึ่งสำหรับวิธีการหาค่าสูงสุดของฟังก์ชันได้

Genetic Algorithm มีแนวคิดมาจากระบบการคัดเลือกสายพันธุ์ตามธรรมชาติ ซึ่งสายพันธุ์ที่ไม่ดีจะค่อย ๆ สูญพันธุ์ไป จนในที่สุดจะเหลือเพียงสายพันธุ์ที่ดีเท่านั้นที่สามารถอยู่รอดสืบต่อไปได้

ขั้นตอน

ขั้นตอนแรกของ Genetic Algorithm จะเริ่มจากการสุ่มข้อมูลขึ้นมาจำนวนหนึ่ง แล้วทำการวัดค่าความดีของข้อมูลแต่ละตัว คัดเลือกเอาเฉพาะข้อมูลที่มีค่าความดีเป็นที่น่าพอใจชุดหนึ่งเก็บไว้ ข้อมูลที่คัดเลือกไว้นั้นจะถูกนำมาทำการผสมพันธุ์กัน ได้เป็นข้อมูลชุดใหม่ ซึ่งเราจะนำข้อมูลชุดใหม่นี้มาทำการคัดเลือกและผสมพันธุ์ต่อไปจนได้ข้อมูลที่มีค่าความดีเป็นที่น่าพอใจ (ดูรูปที่ 1)

การคัดเลือก

การคัดเลือกสายพันธุ์ ทำได้โดยการวัดค่าความดีของข้อมูลแต่ละตัวโดยวิธีใดวิธีหนึ่ง แล้วคัดเอาข้อมูลส่วนหนึ่งที่มีค่าความดีเป็นที่น่าพอใจเก็บไว้ อาจจะคัดเลือกเอาเฉพาะข้อมูลที่มีค่าความดีสูงสุด หรือเลือกข้อมูลที่มีค่าความดีปานกลางและต่ำบางส่วนเข้ามาด้วย เพราะในบางกรณีการนำสายพันธุ์ที่มีค่าความดีปานกลางหรือต่ำมาผสมกันสามารถให้สายพันธุ์ที่ดีในรุ่นต่อไปได้

การผสมพันธุ์

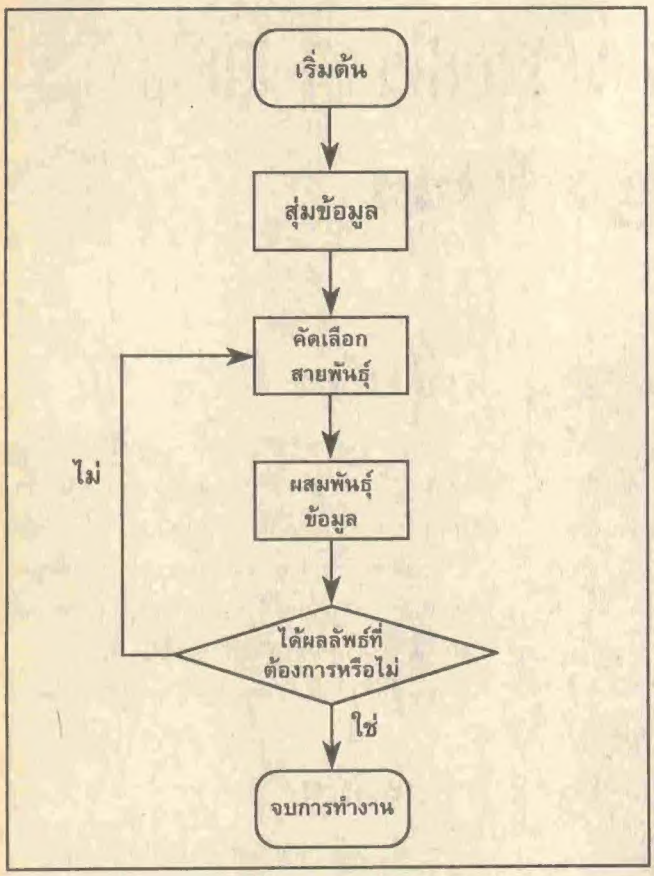
เมื่อได้ข้อมูลที่มีค่าความดีเป็นที่น่าพอใจแล้ว ข้อมูลแต่ละตัวจะถูกนำมาจับคู่กันเพื่อสร้างข้อมูลใหม่ ในการสร้างข้อมูลใหม่เราสามารถทำได้ 2 วิธี คือ การทำ Crossing Over และ Mutation

สำหรับการทำ Crossing Over สามารถทำได้โดยคาดการณ์หาความสัมพันธ์ของข้อมูลทั้ง 2 ที่สามารถทำให้เกิดข้อมูลที่ดีกว่าได้ ซึ่งข้อมูลใหม่ที่ได้จะมีส่วนประกอบที่มาจากข้อมูลเดิมทั้ง 2 ตัว

การทำ Mutation เป็นลักษณะของการผ่าเหล่า คือเราจะนำข้อมูลเก่ามาสุ่มแก้ไขบางส่วนของข้อมูล เช่น Bit บาง Bit ให้เปลี่ยนไป ทำให้ได้ข้อมูลใหม่ที่มีสายพันธุ์ต่างจากเดิม ซึ่งมีโอกาสที่จะเป็นข้อที่ดีขึ้นได้ หรือเราอาจจะสุ่มข้อมูลขึ้นมาใหม่บ้างเพราะในบางครั้งข้อมูลที่ดีที่สุดอาจจะไม่ได้อยู่ในกลุ่มของข้อมูลที่เราสุ่มขึ้นมาในครั้งแรกก็ได้

การหาค่าสูงสุดของฟังก์ชัน

สมมติว่าเรามีฟังก์ชัน $f(x) = x^2 - x$, $a < x < b$ การหาค่าสูงสุดของฟังก์ชันก็คือ การหาค่า x ที่มีค่าอยู่ระหว่าง a และ b ซึ่งเมื่อนำเข้ามาแทนในสมการ $f(x) = x^2 - x$ แล้วทำให้ f(x) มีค่าสูงสุด



รูปที่ 1

โดยปกติเราสามารถใช้วิธีการทางคณิตศาสตร์มาใช้ในการหาค่าสูงสุดของฟังก์ชันได้ แต่บางครั้งการนำวิธีการทางคณิตศาสตร์มาช่วยแก้ปัญหาอาจจะต้องใช้เวลานานมาก Genetic Algorithm จึงเป็นทางเลือกหนึ่งในการแก้ปัญหาเหล่านี้

นำ Genetic Algorithm มาใช้ในการหาค่าสูงสุดของฟังก์ชัน

การสุ่มข้อมูล

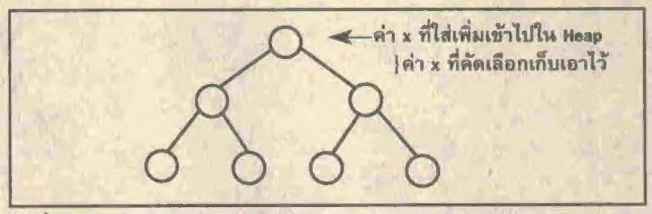
ในการสุ่มค่า x เพื่อหาค่า f(x) ที่มีค่าสูงสุด ควรจะสุ่มให้ค่า x แต่ละตัวมีการกระจายสุงสุด เพื่อให้มีโอกาสพบค่าสูงสุดของฟังก์ชันได้มากขึ้น

สำหรับฟังก์ชันที่เป็นฟังก์ชันเพิ่มและฟังก์ชันลด ค่าสูงสุดของฟังก์ชันจะพบได้เมื่อค่า x เป็นค่ามากที่สุดหรือน้อยที่สุดในช่วงกำหนดไว้ ดังนั้นนอกจากเราจะทำการสุ่มค่า x ขึ้นมาแล้ว เราควรจะเก็บค่า x ที่มีค่ามากที่สุดและน้อยที่สุดในช่วงที่กำหนดไว้ด้วย

การคัดเลือกข้อมูล

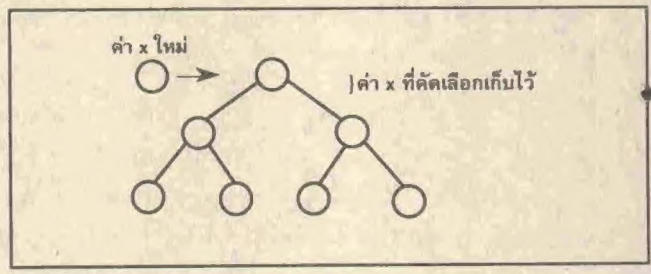
การคัดเลือกค่า x มาจำนวนหนึ่งจากค่า x ทั้งหมด อาจทำได้โดยการจัดเรียงค่า x แต่ละตัวตามค่า f(x) แล้วเลือกเอาค่ามากที่สุด หรืออาจจะเลือกสุ่มเอาค่า x ที่ทำให้ f(x) มีค่าปานกลาง หรือต่ำไปบางส่วนด้วย

หากเราต้องการคัดเลือกเพียงค่า x ที่ทำให้ค่า x ที่ทำให้ค่า f(x) สูงที่สุด โดยไม่เก็บค่า x ที่ให้ค่า f(x) ปานกลางและต่ำเลย เราสามารถกรองเอาเฉพาะค่า x ที่ให้ค่า f(x) สูงสดโดยใช้ Heap ช่วย ซึ่ง Heap ที่ใช้เก็บค่า x สามารถใช้ได้ทั้ง Heap min และ Heap max ดังรูปที่ 2



รูปที่ 2 การเก็บค่า x ใน Heap min

เราจะใช้ Heap min ในการเก็บค่า x โดยค่า x ที่ได้มาใหม่จะถูกใส่เข้าทาง node บนเสมอ เมื่อเราทำ Heapify กับ node บนสุดของ Heap ตามค่า f(x) ค่า x จะถูกเลื่อนลงไปจนอยู่ในตำแหน่งที่ลูกทั้งสองตัวให้ค่า f(x) มากกว่ามัน จะทำให้ค่าของ x ที่ให้ค่า f(x) น้อยที่สุดถูกเลื่อนขึ้นมาที่ node บนสุด ซึ่งค่าใน node บนสุดนี้จะถูกแทนด้วยค่าของ x ที่ใส่เข้ามาใหม่ต่อไป ส่วนค่า x ที่เราจะนำไปผ่านขั้นตอนการ Crossing Over หรือ Mutation นั้นเราจะเอาเฉพาะ Node ที่อยู่ถัดจาก Node บนสุดลงมา (ไม่เอา Node บนสุด)



รูปที่ 3 การเก็บค่า x ใน Heap max

ในการเก็บค่า x ใน Heap max เราจะทำการ Heapify กับค่า x โดยค่า x จะถูกเปรียบเทียบกับ node แต่ละตัว และถูกเลื่อนไปทางค่า node ลงที่น้อยที่สุด ถ้าค่า x ใน node ใดให้ค่า f(x) น้อยกว่า x ที่ใส่เข้าไปใหม่ เราจะใส่ค่า x ลงไปตามตำแหน่งนั้น แล้วใช้ค่า x ที่ถูกแทนที่ทำการ Heapify ต่อไปจนสิ้นสุดขอบเขตของ Heap ค่า x ที่ให้ค่า f(x) น้อยที่สุด จะถูกนำออกมาจาก Heap ส่วนข้อมูลทั้งหมดใน Heap ก็จะถูกนำไป Crossing Over หรือ Mutation ต่อไป

Crossing Over & Mutation

การ Crossing Over ก็คือ การนำข้อมูล 2 ตัว มาทำการตามขั้นตอนต่าง ๆ ซึ่งจะให้ค่า x ต่อใหม่ที่จะเอาไปใช้ในการคัดเลือกครั้งถัดไป ถ้าจะพูดง่าย ๆ การทำ Crossing Over ก็คือ การนำค่า x สองตัวมาผสมกันเพื่อให้ได้ค่า x ตัวใหม่เกิดขึ้นนั่นเอง

การทำ Crossing Over กับค่า x สองตัว สามารถทำได้หลายวิธี เช่น ถ้าเราตั้งสมมติฐานว่าอาจจะมีค่า x ที่มีค่าอยู่ระหว่างค่า x สองตัวที่ทำให้ f(x) มีค่าสูงสุดได้ การ Crossing Over ก็อาจทำได้โดยการนำค่า x ทั้งสองตัวมาหาค่ากลาง ($(x_1 + x_2) / 2$) หากเรามองค่า x ทั้งสองตัวเป็นสายของ Bit (ลักษณะเดียวกับโครโมโซม) แล้วเลือก Bit มาจากค่า x ทั้งสองตัวทีละ Bit หรือมองค่า x เป็นสายของข้อมูลที่แต่ละตัวมีขนาด n Bit แล้วเลือกเอาข้อมูล n Bit ของค่า x ทั้งสองตัวที่ตำแหน่งเดียวกันตัวใดตัวหนึ่งมาประกอบกันเป็นค่า x ตัวใหม่

นอกจากวิธีที่ผมยกตัวอย่างมาแล้ว เรายังมีวิธีสร้างค่า x ใหม่ได้อีกหลายวิธี ทั้งการกำหนดเพศให้กับค่า x แต่ละตัว และวิธีอื่น ๆ อีกมากมาย เพื่อให้ได้ค่า x ตัวใหม่ที่สามารถให้ค่า f(x) มากกว่าค่า f(x) ที่ได้จากค่า x เดิมทั้งสองตัว แต่บางครั้งค่า x ที่ให้ค่า f(x) สูงสุด ก็อาจจะไม่อยู่ในกลุ่มของค่า x ที่สุ่มขึ้นมา และไม่สามารถทำ Crossing Over ให้เกิดค่า x ที่ให้ค่า f(x) ที่มีค่าสูงสุดได้ ดังนั้นหลังจากทำ Crossing Over หลาย ๆ ครั้ง เราก็จะสมมติให้เกิดการผ่าเหล่าขึ้น โดยสุ่มแก้ไข Bit บาง Bit ของค่า x ให้มีค่าเปลี่ยนแปลงไปจากเดิม หรืออาจจะสุ่มค่า x ขึ้นมาใหม่โดยไม่มีความสัมพันธ์กับค่า x เดิมเลยก็ได้ ขั้นตอนนี้ถูกเรียกว่า Mutation ซึ่งค่า x ที่ได้จากการทำ Mutation นี้อาจจะให้ค่า x ที่ให้ค่า f(x) สูงกว่าเดิมหรือต่ำกว่าเดิมก็ได้ ถ้าค่า x ใหม่ที่ได้เลวลง (ทำให้ f(x) มีค่าน้อยกว่าจากค่า x เดิม) ค่า x ที่ได้นั้นก็จะถูกคัดออกไปในขั้นตอนการคัดเลือกเอง

การตรวจสอบคำตอบ

การตรวจสอบว่าค่า x ที่ได้เป็นค่าที่ทำให้ f(x) มีค่าสูงสุดหรือไม่ สามารถตรวจสอบได้โดยการทำตามขั้นตอนของ Genetic Algorithm จนกระทั่งค่า x ที่ถูกคัดเลือกเป็นค่าเดิมตลอด หรือกำหนดจำนวน Generation ที่จะทำการค้นหาไว้ก่อน เมื่อถึง Generation ที่กำหนด ให้แสดงค่า x ที่ทำให้ f(x) มีค่าสูงสุดในกลุ่มของค่า x ที่คัดเลือกไว้

สำหรับผู้ที่กำลังศึกษาและมีความสนใจในเรื่องของ Genetic Algorithm ผมได้จัดเตรียมตัวอย่างการเขียนโปรแกรมไว้ให้แล้ว หวังเป็นอย่างยิ่งว่าจะเป็นประโยชน์สำหรับท่าน ซึ่งสามารถขอบริการได้จากซอฟต์เมล์

เลื่อนจอภาพใน Microsoft Office โดยไม่ต้องใช้ Scrollbar

ผู้ที่ใช้งานโปรแกรมชุด Microsoft Office 97 ไม่ว่าจะเป็น Microsoft Word 97, Microsoft Excel 97 หรือ Microsoft Access เวลาต้องการตรวจสอบงานที่ได้พิมพ์ไปหลายครั้งจะต้องมีการใช้แถบเลื่อน (Scrollbar) เพื่อตรวจสอบงาน ซึ่งต้องเสียเวลาในการตรวจสอบ และหน้าจอที่แสดงเอกสารเวลาเลื่อนอาจจะไม่นิ่มนวลเท่าที่ควร มีเทคนิคที่จะเลื่อนจอภาพได้อย่างนิ่มนวลและสามารถปรับให้เร็วหรือช้าได้โดยไม่ต้องใช้แถบเลื่อนหรือกดคีย์บอร์ดเลย

เทคนิคที่ว่านี้คือ **ผู้ใช้จะต้องใช้เมาส์แบบ 3 ปุ่ม** ปุ่มกลางของเมาส์จะมีประโยชน์ในการเลื่อนภาพโดยอัตโนมัติ เวลาใช้งานเพียงแค่กด (click) เมาส์ปุ่มกลาง แล้วเลื่อนเมาส์ไปตามทิศทางที่ต้องการ (ขึ้น-ลง, ซ้าย-ขวา) ก็จะปรากฏภาพลูกศรซึ่งแสดงทิศทางของการเลื่อน ถ้าใช้โปรแกรม Microsoft Excel 97 สามารถควบคุมทิศทางการเลื่อนได้ทั้งสี่ทิศ หากต้องการเปลี่ยนทิศทางก็เพียงแต่เลื่อนเมาส์ไปตามทิศทางที่ต้องการ หลังจากนั้นก็ไม่จำเป็นต้องจับเมาส์อีก เพราะภาพเอกสารจะถูกเลื่อนไปโดยอัตโนมัติ และสามารถควบคุมความเร็วของการเลื่อนได้

ในโปรแกรม Microsoft Word 97 สามารถเลื่อนภาพเอกสารได้เพียงสองทางเท่านั้นคือ เลื่อนขึ้น หรือเลื่อนลง ซึ่งโปรแกรมนี้สามารถควบคุมความเร็วของการเลื่อนภาพเอกสารแต่ละหน้าได้ ถ้าต้องการเพิ่มความเร็วในการเลื่อนก็เพียงแต่เลื่อนภาพลูกศรที่ปรากฏในแถบเลื่อนทางด้านขวาให้ห่างออกจากจุดกึ่งกลางของแถบเลื่อนให้มากขึ้น ภาพก็จะเลื่อนได้เร็วขึ้น ในทำนองเดียวกัน ถ้าต้องการหยุดการเลื่อนหรือเลื่อนภาพให้ช้าลง ทำได้โดยการเลื่อนภาพให้ใกล้กับจุดกึ่งกลางของแถบเลื่อน ถ้าเข้าใกล้จุดกึ่งกลางของแถบเลื่อนมาก ๆ จะทำให้ภาพหยุดเลื่อนได้ แต่ยังอยู่ในโหมดของการเลื่อน หากต้องการยกเลิกการเลื่อนภาพเอกสารโดยอัตโนมัตินี้ ทำได้โดยการกด (click) เมาส์ปุ่มใดก็ได้อีกครั้ง ก็จะกลับสู่โหมดปกติ

ในโปรแกรม Microsoft Access 97 ก็สามารถเลื่อนได้แต่จะเป็นแบบกึ่งอัตโนมัติ โดยวิธีนี้จะต้องกดเมาส์ปุ่มกลางค้างไว้ตลอดจึงจะสามารถเลื่อนภาพไปในทิศทางที่ต้องการได้ ถ้าไม่กดก็จะไม่สามารถใช้งานในโหมดการเลื่อนภาพได้

ในโปรแกรม Microsoft Powerpoint 97 เท่าที่ลองทดสอบ ปรากฏว่าไม่สามารถเลื่อนภาพด้วยวิธีการนี้ได้

อาจารย์กฤษดา ยิ่งขยัน
แผนกอิเล็กทรอนิกส์ คณะวิชาช่างไฟฟ้า
สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคพายัพ ถ.ห้วยแก้ว อ.เมือง จ.เชียงใหม่ 50300

เจ้าของเทคนิคจะได้รับโปรแกรม Thaisoft Dictionary v2.0 จำนวน 1 ชุด มูลค่า 1,485 บาท